What is claimed is:

1. A data model representing semantic information associated with objects stored in a file system, the data model comprising:

a first object identifier identifying a first object stored in the file system;

a second object identifier identifying a second object stored in the file system, the second object being related to the first object; and

a relation identifier identifying a relationship between the first object and the second object.

2. The data model of claim 1, wherein the first object comprises a first file stored in the file system.

3. The data model of claim 2, wherein the second object comprises one or more of a second file generated from the first file and meta data generated from the first file.

4. The data model of claim 3, wherein the relation identifier is a semantic of the first file.

5. The data model of claim 1, wherein the data model includes a tuple in a format and order comprising:

the first object identifier, the relation identifier, the second object identifier.

6. The data model of claim 1, wherein the relation identifier comprises a property of the first object and the second object comprises a value of the property for the first object.

7. The data model of claim 1, wherein the data model represents a function operable to be performed in the semantic file system.

8. The data model of claim 7, wherein the function is associated with one or more of generating a view of the objects stored in the semantic file system, restricting access to an object in the file system, searching in the semantic file system, performing an action based on at least one predetermined condition, and performing archival functions in the semantic file system.

9. The data model of claim 1, wherein the relation identifier identifies a dependency between the first object and the second object.

10. The data model of claim 9, wherein the dependency is associated with version information for the first object.

11. The data model of claim 9, wherein the dependency is associated with a hierarchal file space.

12. The data model of claim 9, wherein the dependency is associated with one or more users or one or more applications.

13. The data model of claim 12, wherein the dependency is used to generate file space views for the one or more users or for the one or more applications.

14. The data model of claim 1, wherein the relation identifier identifies the second object as including property semantic information for the first object, the property semantic information including one or more of statistical information for the first object and a description of the first object.

15. The data model of claim 1, wherein the relation identifier identifies the second object as including context semantic information for the first object, the context semantic information being associated with access patterns for the first object.

16. The data model of claim 15, wherein the first object is a file and the access patterns are associated with one or more other files accessed before or after the file.

17. The data model of claim 1, wherein the relation identifier identifies the second object as including content-based semantic information associated with contents of the first object.

18. The data model of claim 1, wherein the data model is used to represent multiple types of relation identifiers in a schema.

19. The data model of claim 18, wherein the schema is modifiable to include a new relation identifier or to remove a relation identifier currently in the schema.

20. A method associated with a file system, the method comprising:

storing objects in the file system including a first object and a second object, wherein the first object is related to the second object; and

storing a relation meta data identifying a relationship between the first object and the second object, wherein the relationship is represented by a data model including a first identifier identifying the first object; a second identifier identifying the second object; and a relation identifier identifying the relationship between the first object and the second object.

21. The method of claim 20, wherein the first object includes a first file stored in the file system and the second object includes one or more of a second file stored in the file system and semantic information for the first file.

22. The method of claim 20, further comprising:

determining whether the first object in the file system is accessed;

identifying a predetermined condition associated with the first object in response to the first object being accessed; and

performing an action in response to the predetermined condition existing, wherein the relation identifier identifies the predetermined condition and the action.

23. The method of claim 20, further comprising:

determining a user-related or application-related dependency between the first object and the second object;

generating a view of the file system based on the dependency.

24. The method of claim 20, further comprising:

executing a query of the stored objects; and

generating a file space view from search results of the executed query.

25. The method of claim 20, further comprising:

generating a schema using a plurality of relation meta data, the plurality of relation meta data identifying relationships between one or more of the objects.

26. The method of claim 25, wherein the schema is modifiable by adding or removing relation meta data from the schema.

27. The method of claim 25, wherein at least one of the plurality of relation meta data is determined through property inheritance for the schema.

28. The method of claim 20, further comprising:

identifying a restriction on accessing the first object from the relation, wherein the first object is a file and the second object identifies one or more of a user and an application having restricted access to the file.

29. The method of claim 20, further comprising:

extracting semantic information for the objects; and

storing the semantic information.

30. The method of claim 29, further comprising:

receiving a request for information stored in the file system; and

searching the semantic information to identify any files stored in the file system that meet the request.

31. The method of claim 30, wherein the semantic information includes one or more types of semantic information comprising content-based semantic information related to the contents of files stored in the file system, context-based semantic information related to user access patterns of the files stored in the file system, and property semantic information related to statistics or descriptions of the files stored in the file system.

32. The method of claim 31, wherein searching the semantic information comprises: searching a plurality of the types of semantic information.

33. The method of claim 31, further comprising:

returning results of the search using a precision variable, wherein the precision variable is related to a relevance of search results to the search request

34. The method of claim 30, further comprising using one or more of a placement algorithm and a caching algorithm for placing or caching related objects in the file system.

35. The method of claim 29, further comprising:

identifying one or more files in the file system to be archived based on the semantic information associated with the one or more files; and

archiving the identified files.

36. A semantic file system comprising:

at least one storage device storing files and semantic information related to the files, wherein

relationships between one or more files stored in the file system and relationships between the files and associated semantic information are represented using a data model including a first identifier identifying a first file of the files; a second identifier identifying a second file of the files or semantic information associated with the first file; and a relation identifier identifying the relationship between the first file and the second file or identifying the relationship between the first file and the semantic information associated with the first file.

37. The semantic file system of claim 36, further comprising:

a repository storing meta data identifying the relationships between the one or more files and the relationships between the files and the semantic information; and

at least one controller operable to access the meta data to perform a function associated with at least one of the files.

38. The semantic file system of claim 37, wherein the function is related to one of versioning, organizing a file space, performing an action on the at least one of the files in response to a predetermined condition existing, searching the semantic file system, and archiving.

39. The semantic file system of claim 36, wherein the semantic file system is a distributed file system.

40. A file system, comprising:

storage means for storing a plurality of files, semantic information for the plurality of files and relation meta data identifying relationships between one or more

of at least some of the plurality of files and between the plurality of files and the semantic information, wherein

a data model represents the relationships and the data model comprises a first object identifier identifying a first object wherein the first object includes a file of the plurality of files, a second object identifier identifying a second object wherein the second object includes one of a second file of the plurality of files and semantic information for the first file, and a relation identifier identifying a relationship of the relationships between the first object and the second object.

41. The system of claim 40, further comprising extraction means for extracting the semantic information from the plurality of files.

42. The system of claim 40, further comprising event means for determining whether a file of the plurality of files is accessed, identifying a predetermined condition associated with the file, and performing an action in response to the predetermined condition existing.

43. The system of claim 40, further comprising view means for generating a view of a file space where the plurality of files are stored based on user-related or application-related dependencies between at least some of the plurality of files.

44. The system of claim 40, further comprising search means for receiving a search request and for searching information stored in the storage means that meets the search request.

45. The system of claim 40 further comprising archiving means for archiving the files stored in the storage means.

5